8Н

УДК 681.3

І.А. Назарова
Донецький національний технічний університет, м. Донецьк, Україна
nazarova@r5.dgtu.donetsk.ua

Підвищення ефективності паралельного розв'язання лінійної задачі Коші на основі методу рекурсивного множення матриць

Запропоновано масштабований паралельний метод матричного добутку на основі систолічного та рекурсивного алгоритмів, який дозволяє підвищити ефективність розв'язання лінійної задачі Коші на основі експоненціального методу. Для розробленого алгоритму визначено оптимальні значення глибини рекурсії і розміру мінімального блоку перемножуваних матриць. Розроблено схеми відображення методу на паралельні структури з розподіленою пам'яттю топології сітка/тор.

Вступ

Дослідження, викладені у статті, присвячені розробці і аналізу ефективності паралельних методів розв'язання задачі Коші для систем лінійних звичайних диференціальних рівнянь (СЛЗДР) із вбудованими засобами оцінки локальної похибки. Зокрема, запропоновані спеціальні методи вирішення однорідних і неоднорідних СЛЗДР з постійними коефіцієнтами. Як показали дослідження, урахування специфіки задачі дозволяє отримати більш ефективні паралельні обчислювальні алгоритми, ніж у випадку стандартних чисельних схем.

Експоненціальний метод відноситься до спеціальних методів чисельного інтегрування початкової задачі Коші для СЛЗДР, заснований на точному представленні розв'язку в аналітичній формі і наближеному обчисленні матричної експоненти [1]. У загальному вигляді задачу Коші для однорідних СЛЗДР з постійними коефіцієнтами можна записати таким чином:

$$\begin{cases} \bar{y}'(x) = A \cdot \bar{y}(x), \\ \bar{y}(x_0) = \bar{y}_0, \ A = const, \end{cases} \qquad (1)$$

де $\bar{y} = (y_1, y_2, ..., y_m)^T$ – вектор невідомих,

$\bar{y}_0 = (y_{10}, y_{20}, ..., y_{m0})^T$ – вектор початкових умов,

$A = \|a_{ij}\|, \ i, j = \overline{1, m}$ – матриця коефіцієнтів лінійної системи, елементи якої є константами.

Точний розв'язок задачі Коші виду (1) потребує обчислення матричної експоненти:

$$\begin{cases} \bar{y}(x_n + h) = e^{hA} \cdot \bar{y}(x_n), \\ e^{hA} = F(hA) = \sum_{i=0}^{\infty} \frac{(hA)^i}{i!}. \end{cases} \qquad (2)$$

Наближений розв’язок на кроці можна побудувати, використовуючи апроксимацію матричної експоненти відрізком ряду Тейлора при малому значенні h:

$$\begin{cases} \bar{y}_{n+1} = F_r(hA) \cdot \bar{y}_n, \\ F_r(hA) = \sum_{i=0}^{r} \frac{(hA)^i}{i!}, \end{cases} \quad (3)$$

а потім застосувати деякий алгоритм множення матриць.

Паралельний блоковий рекурсивно-систолічний метод матричного добутку

Матричний добуток (МД) – домінуюча обчислювальна частина експоненціальних методів розв’язання СЗДР із вбудованими засобами оцінки локальної похибки. Тому прискорення паралельної реалізації цієї базової операції лінійної алгебри означає зменшення терміну виконання методу вирішення лінійної задачі у цілому. При величезній різноманітності методів обчислення матричного добутку [2-5] для щільно заповнених матриць є два принципово різних класи послідовних алгоритмів: традиційні і рекурсивні методи на базі швидкого множення Штрассена [6-8]. У оригіналі алгоритм Штрассена – це алгоритм множення блокових матриць половинного розміру, де кожен блок квадратний, тобто розмірності матриць мають бути парними числами. Метод Штрассена – Винограда [7] складається з 7 блокових множень і 15 блокових складань\віднімань матриць (рис. 1).

$$\begin{array}{lll}
S_1 = A_{21} + A_{22}, & M_1 = S_2 S_6, & T_1 = M_1 + M_3, \\
S_2 = S_1 - A_{11}, & M_2 = A_{11} B_{11}, & T_2 = T_1 + M_4, \\
S_3 = A_{21} - A_{12}, & M_3 = A_{12} B_{21}, & T_3 = M_5 + M_6, \\
S_4 = A_{12} - S_2, & M_4 = S_3 S_7, & C_{11} = M_2 + M_3, \\
S_5 = B_{12} - B_{11}, & M_5 = S_1 S_5, & C_{12} = T_1 + T_3, \\
S_6 = B_{22} - S_5, & M_6 = S_4 B_{22}, & C_{21} = T_2 - M_7, \\
S_7 = B_{22} - B_{12}, & M_7 = A_{22} S_8, & C_{22} = T_2 + M_5, \\
S_8 = S_6 - B_{12}, & &
\end{array}$$

$$A = \left\langle \frac{A_{11}}{A_{21}} \middle| \frac{A_{12}}{A_{22}} \right\rangle, \quad B = \left\langle \frac{B_{11}}{B_{21}} \middle| \frac{B_{12}}{B_{22}} \right\rangle, \quad C = \left\langle \frac{C_{11}}{C_{21}} \middle| \frac{C_{12}}{C_{22}} \right\rangle.$$

Рисунок 1 – Метод швидкого множення матриць Штрассена – Винограда

Ідея Штрассена може бути застосована рекурсивно для знаходження добутків блоків матриць $M_i, i = \overline{1,7}$. Якщо початкові матриці A і B порядку m, то алгоритм швидкого множення можна використовувати багато разів, отримуючи на найнижчому рівні рекурсії блоки $k \times k = 1 \times 1$. Проте немає необхідності опускатися униз до рівня блоків одиничного порядку. При досить малих розмірах блоку ($k \le k_{min}$) може виявитися корисним обчислювати блоки, використовуючи стандартний алгоритм МД. На першому кроці алгоритм передбачає 7 звернень до самого себе з матрицями порядку $m/2$ і 15 операцій типу складання матриць того ж порядку. Далі йде розгортка рекурсії до досягнення мінімального розміру блоку і множення блоків за традиційним алгоритмом матричного множення. Обчислювальна складність запропонованої схеми алгоритму швидкого множення визначається функцією розмірності початкових матриць і мінімального порядку перемножуваних блоків: m, k_{min}.

Нехай при виконанні матричного множення алгоритм Штрассена – Винограда рекурсивно викликався d раз, тоді порядок блоків матриць, що множаться, дорівнює $k_{min} = m/2^d$. Час реалізації послідовного алгоритму при цьому складає:

$$T_1^{Str}(m) = \left[2\left(\frac{7}{8}\right)^d m^3 + \frac{15}{4} m^2 \left(1 + \frac{7}{4} + \frac{7^2}{4^2} + \ldots + \frac{7^{d-1}}{4^{d-1}} \right) \right] \cdot t_{op}.$$

Відомий паралельний алгоритм класичного методу Штрассена був застосований на *Intel Paragon* і показав добрі результати порівняно з традиційними алгоритмами МД [6-7]. Істотним недоліком цього алгоритму є відсутність масштабованості, оскільки необхідне число процесорів для нього кратно сімці (по кількості множень блоків матриць $M_i, i = \overline{1,7}$). Це обмеження є достатньо жорстким і не природним для більшості паралельних архітектур.

Для подолання вказаного недоліку скористаємося поліалгоритмічним підходом. Під поліалгоритмом мається на увазі комбінація двох або більш алгоритмів в одну обчислювальну схему, що реалізує поставлену задачу з метою скорочення обчислювальних і/або ємкісних витрат. Алгоритм швидкого множення рекурсивний, тому є можливість побудувати поліалгоритм з деякого традиційного алгоритму множення матриць на верхньому рівні рекурсії і методу Штрассена – Винограда на нижньому рівні, і навпаки. Алгоритм Штрассена є блоковим, тому природньо комбінувати його з алгоритмом матричного добутку, що використовує відповідне розбиття даних. Побудуємо поліалгоритм з блокового систолічного алгоритму матричного множення між процесорами і серії вживань рекурсивного методу Штрассена на кожному процесорі.

Нехай є замкнута двовимірна сітка процесорів розмірності $p \times p$, початкові матриці A і B розподілені на блоки $k = m/p$, кількість блоків дорівнює $q \times q = p \times p$. Блоки початкових матриць і результату з координатами $< i, j >$ зберігаються у відповідному процесорі з тими ж координатами. Як і раніше, передбачаємо, що $m \ mod \ p = 0$, інакше використовується доповнення нульовими елементами. Спочатку, за обчислювальною схемою блокового систолічного множення, виконується косе зрушення вліво по рядках для блоків матриці A і косе зрушення вгору по стовпцях для блоків матриці B. На кожному з p кроків алгоритму проводиться множення блоків матриць A і B, що зберігаються в процесорі з номером, і складання з вже обчисленим значенням блоку матриці C_{ij}, розташованим на цьому ж процесорі. Для першого кроку це значення дорівнює нульовому блоку. Потім проводиться одиночне зрушення вліво по рядках паралельно для всіх блоків матриці $A: A_{ij} \leftarrow A_{i,j+1}$ і одиночне зрушення вгору по стовпцях також паралельно для всіх блоків матриці $B: B_{ij} \leftarrow B_{i+1,j}$. Множення блоків матриць виконується усередині одного процесора за рекурсивним алгоритмом, що дозволяє уникнути додаткових пересилок даних. На нижньому рівні рекурсії застосовується стандартний алгоритм множення матриць, глибина рекурсії дорівнює d. Розроблений алгоритм є таким, що масштабується для будь-якого числа процесорів і будь-якого порядку матриць, що множаться. Час реалізації блокового рекурсивно-систолічного алгоритму включає час виконання арифметичних і обмінних операцій:

$$T_p^{BSys-Str} = T_{p,comp}^{BSys-Str} + T_{p,comm}^{BSys-Str}.$$

При цьому час виконання обчислень за схемою дорівнює:

$$T_{p,comp}^{BSys-Str} = q \cdot \left(T_{A_{ij} \underset{Str}{\times} B_{ij}} + T_{A_{ij}+B_{ij}} \right),$$

де $T_{A_{ij} \underset{Str}{\times} B_{ij}}$ – час множення блоків матриць порядку $k=m/p$ за рекурсивним алгоритмом; $T_{A_{ij}+B_{ij}} = k^2 \cdot t_{ad} = \left(m^2/p^2\right)\cdot t_{ad}$ – час складання блоків матриць тієї же розмірності.

Час виконання множення блоків виконується за алгоритмом Штрассена – Винограда і задовольняє рекурентному співвідношенню:

$$T_{A_{ij} \underset{Str}{\times} B_{ij}} = T_p^{Str}(k) = 7T_p^{Str}\left(\frac{k}{2}\right) + 15\left(\frac{k}{2}\right)^2 \cdot t_{ad}.$$

У свою чергу, час реалізації алгоритму для матриць порядку $k/2,..., k/2^{d-1}$ відповідно дорівнює:

$$T_p^{Str}\left(\frac{k}{2}\right) = 7T_p^{Str}\left(\frac{k}{4}\right) + 15\left(\frac{k}{4}\right)^2 t_{ad},$$

$$\ldots,$$

$$T_p^{Str}\left(\frac{k}{2^{d-1}}\right) = 7T_p^{Str}\left(\frac{k}{2^d}\right) + 15\left(\frac{k}{2^d}\right)^2 t_{ad}.$$

Виконавши підстановки і елементарні перетворення, отримаємо:

$$T_p^{Str}(k) = 7^d T_p^{Str}\left(\frac{k}{2^d}\right) + \frac{15}{4}k^2\left(1 + \frac{7}{4} + \frac{7^2}{4^2} + \ldots + \frac{7^{d-1}}{4^{d-1}}\right)t_{ad}.$$

До цих пір була визначена обчислювальна складність розгортки рекурсії. Розглянемо внутрішній рівень рекурсії, оскільки саме там виконуються операції множення блоків матриць мінімального порядку за стандартним методом матричного множення:

$$T_p^{Str}\left(\frac{k}{2^d}\right) = k_{min}^3 \cdot \left(t_{mul} + t_{ad}\right) = 2\left(\frac{k}{2^d}\right)^3 \cdot t_{op}.$$

Таким чином, час реалізації швидкого рекурсивного множення блоків матриць задля мультикомп'ютера складає:

$$T_p^{Str}(k) = \left[2\left(\frac{7}{8}\right)^d k^3 + \frac{15}{4}k^2\left(\frac{1-\frac{7^d}{4^d}}{1-\frac{7}{4}}\right)\right]\cdot t_{op}.$$

Тоді загальний час виконання арифметичних операцій для комбінації блокового систолічного алгоритму і рекурсивного матричного множення дорівнює:

$$T_{p,comp}^{BSys\text{-}Str} = \left[2\left(\frac{7}{8}\right)^d \frac{m^3}{p^2} + \left(5\left(\frac{7}{4}\right)^d - 4\right)\frac{m^2}{p}\right]\cdot t_{op}.$$

Час обмінних операцій для описаної схеми визначається, як і для блокового систолічного алгоритму:

$$T_{p,comm}^{BSys-Str} = 4(p-1)\cdot\left(t_s + \frac{m^2}{p^2}\cdot t_w\right).$$

Очевидно, що динамічні характеристики паралельних обчислювальних схем матричного добутку залежать від співвідношення між числом процесорів і розмірністю матриць. Для рекурсивного алгоритму множення матриць істотним параметром є також величина глибини рекурсії, d (рис. 2 – 4).

```
(* Визначення оптимального значення глибини рекурсії *)

Round[Simplify[Log[2, 2*m*(Log[8/7])] - Log[2, (5*Log[7/4])]]]
```

$$d_{opt} = \text{Round}\left[\frac{\text{Log}[2\,m\,\text{Log}[\frac{8}{7}]] - \text{Log}[5\,\text{Log}[\frac{7}{4}]]}{\text{Log}[2]}\right]$$

$$m_{opt} = \text{N}\left[m / 2^{d_{opt}}\right] = 8$$

Рисунок 2 – Визначення оптимального значення глибини рекурсії для рекурсивного алгоритму МД

Визначення оптимальної величини глибини рекурсії, а отже, і розмірності мінімального оброблюваного блоку матриць, що множаться, проводилося стандартним математичним апаратом в пакеті *Mathematica*. Для цього необхідно знайти значення, що доставляє мінімум функції T_1^{Str}, причому діапазон зміни глибини рекурсії обмежений наступною нерівністю: $k_{min} = \left(m / 2^d\right) \geq 1 \Rightarrow m \geq 2^d \Rightarrow d \leq log_2 m$. Очевидно, що при великих розмірностях СЛЗДР кожна з функцій $V(d)=T_1^{Str}/t_{op}$ рис. 3 має явно виражений локальний мінімум, відношення ж розмірності системи до глибини рекурсії при цьому залишається постійним. Наприклад, при $m = 1024$, $d_{min} = 7$ і $m / 2^{d_{min}} = 1024 / 128 = 8$, а при $m = 256$ маємо $d_{min} = 5 \Rightarrow m / 2^{d_{min}} = 8$. У той же час при $m = 16$ мінімальна глибина рекурсії дорівнює 1, тобто для досягнення мінімуму обчислювальних витрат необхідно виконати всього один крок розгортки рекурсії, для $m = 8$ розгортки рекурсії немає, оскільки $d_{min} = 0$.

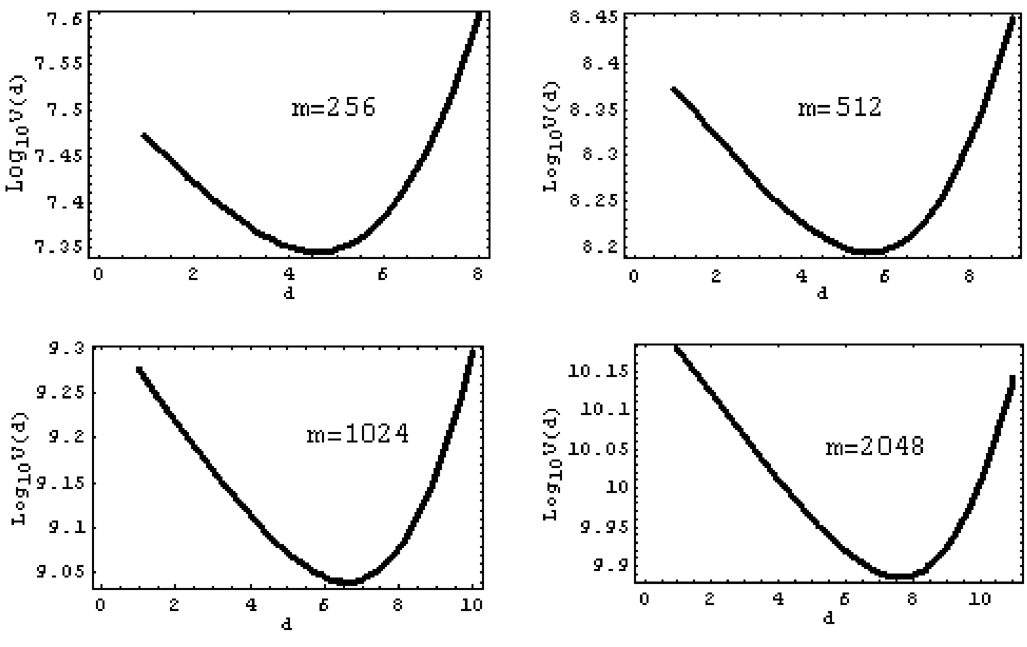


Рисунок 3 – Обчислювальна складність рекурсивно-систолічного алгоритму від глибини рекурсії у логарифмічному масштабі

Аналітичні результати підтверджуються результатами експериментів, на рис. 4 наведено залежність часу виконання матричного добутку від величини блоку матриць, що перемножуються.

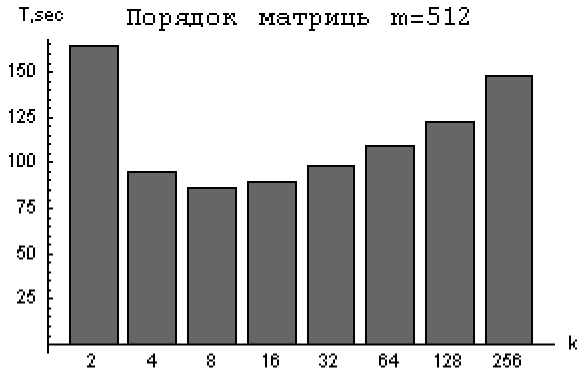


Рисунок 4 – Експериментальне визначення величини мінімального блоку для рекурсивно-систолічного алгоритму множення матриць

Для паралельного варіанта запропонованого алгоритму блокового рекурсивно-систолічного множення окрім глибини рекурсії необхідно врахувати співвідношення між числом процесорів, p і розмірністю матриць, m (рис. 5). Обмеження на діапазон зміни глибини рекурсії визначаються наступною нерівністю: $k_{min} = \frac{m}{2^d p} \geq 1 \Rightarrow \frac{m}{p} \geq 2^d \Rightarrow d \leq log_2 \frac{m}{p}$. При оцінюванні глибини рекурсії часові характеристики наведені до часу реалізації однієї операції з плаваючою точкою, для різних мультикомп'ютерів величина оптимальної глибини рекурсії має бути поправлена з врахуванням цієї машинно-залежної константи.

```
(* Визначення оптимального значення глибини рекурсії *)

pd_opt = Round[Simplify[Log[2, 2*m/p*(Log[8/7])] - Log[2, (5*Log[7/4])]]]
```

$$pd_{opt} = Round\left[\frac{Log\left[\frac{2\, m\, Log\left[\frac{8}{7}\right]}{p}\right] - Log\left[5\, Log\left[\frac{7}{4}\right]\right]}{Log[2]}\right]$$

$$(m/p)_{opt} = 8 * 2^{pd_{opt}}$$

Рисунок 5 – Визначення величини оптимальної глибини рекурсії для паралельного блокового рекурсивно-систолічного алгоритму

Аналіз аналітичних виразів, що характеризують динамічні характеристики якості паралельних алгоритмів, а також проведений чисельний експеримент, дозволяють зробити наступні висновки:

1) паралельний блоковий метод рекурсивно-систолічного множення матриць володіє меншою асимптотичною $T_p^{bl-sys} / T_p^{Bsys-Str} \approx (8/7)^d$ і реальною, підтверджуваною експериментальним шляхом, тимчасовою складністю порівняно з найбільш відомими традиційними паралельними алгоритмами множення матриць (рис. 6);

2) динамічні характеристики запропонованого алгоритму на основі швидкого множення перевершують відповідні характеристики стандартних аналогів, особливо для

матриць великих розмірностей, при цьому коефіцієнти прискорення і ефективності зростають із зростанням розмірності задачі: $m\uparrow \Rightarrow S\uparrow \Rightarrow E\uparrow$ і зменшуються із зростанням числа процесорів $p\uparrow \Rightarrow S\downarrow \Rightarrow E\downarrow$ (рис. 7).

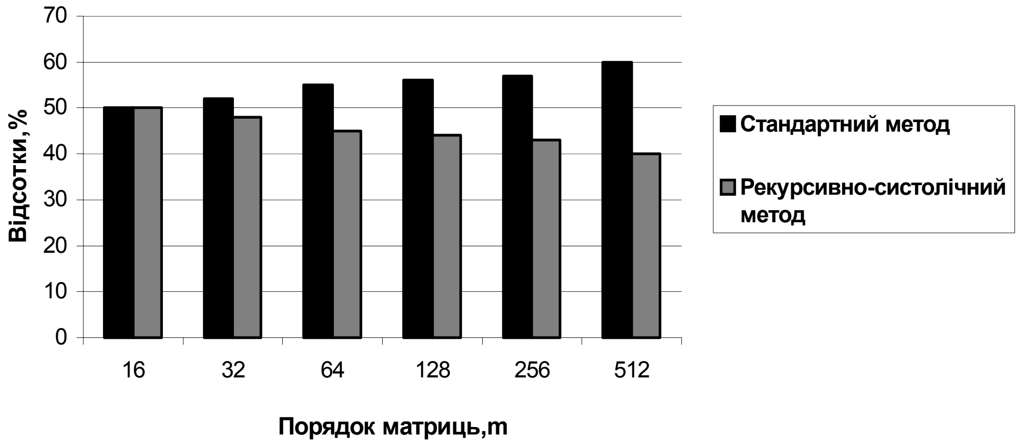


Рисунок 6 – Співвідношення тривалостей реалізації паралельних методів стандартного і рекурсивно-систолічного матричного добутку

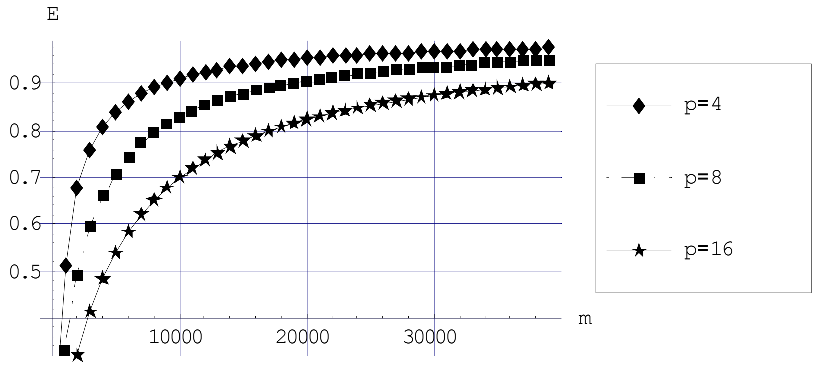


Рисунок 7 – Коефіцієнти ефективності паралельного блокового рекурсивно-систолічного методу для мультикомп'ютерів з шириною процесорної сітки, рівною p

Крім того, рекурсивний характер розробленого алгоритму дозволяє звести багатовимірну початкову задачу до вирішення підзадач меншої розмірності і, за рахунок використання швидкої пам'яті комп'ютера, досягти прискорення операції матричного добутку. До недоліків цього підходу слід віднести дещо гіршу чисельну стійкість, хоча і достатню для більшості практичних задач [9-10].

Висновки

Чисельний експеримент на основі тестів для СЗДР та проведений порівняльний аналіз динамічних характеристик паралельних методів розв'язання лінійної задачі Коші продемонстрував, що запропоновані експоненціальні методи мають меншу тимчасову складність порівняно зі стандартними аналогами. Крім того, розроблений паралельний алгоритм на основі комбінації швидкого рекурсивного і систолічного алгоритмів матрич-

ного добутку дозволив підвищити ефективність паралельного розв'язання лінійної задачі, як для MIMD, так і для кластерних архітектур з розподіленою пам'яттю та топологією двовимірний тор. Використання розробленого рекурсивно-систолічного методу дозволило в $(8/7)^d$, $d = 1,2...$ раз прискорити виконання цієї найбільш ресурсоємної операції при вирішенні лінійних СЗДР на основі експоненти.

Перспективним напрямком дослідження є застосування рекурсивно-систолічного блокового методу множення матриць у розв'язанні нелінійної задачі Коші для блокових багатоточкових неявних методів, що дозволить підвищити ефективність вирішення жорстких динамічних задач.

Література

1. Иванов В.В. Методы вычислений на ЭВМ. – К.: Наукова думка, 1986. – 584 с.
2. Голуб Д., Ван Лоун Ч. Матричные вычисления: Пер. с англ. – М.: Мир, 1999. – 548 с.
3. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения: Пер. с англ. – М.: Мир, 2001. – 430 с.
4. Demmel J., Higham N.J. Stability of block algorithms with fast Level 3 BLAS // ACM Trans. Math. Software. – 1992. – № 18. – P. 274-291.
5. Choi J., Dongarra J., Walker D. PUMMA: Parallel universal matrix multiplication algorithms on distributed memory concurrent computers. – Режим доступа: http://citeseer.ist.psu.edu/choi93pumma.html
6. Strassen V. Gaussian elimination is not optimal // Numer. Math. 13. – P. 354-356.
7. Pan V. How can we speed up matrix multiplication // SIAM Rev. – 1984. – № 26. – P. 393-416.
8. Winograd S. A new algorithm for inner product // IEEE Trans. Comp. C-17. – P. 693-694.
9. Bailey D.H. Extra high speed matrix multiplication on CRAY-2 // SIAM J. Sci. and Stat. Comp. 9. – P. 603-607.
10. Bailey D.H., Lee K., Simon H.D. Using Strassen's algorithm to accelerate the solution of linear systems // J. Supercomputing. – 1991. – № 4. – P. 97-371.

И.А. Назарова

Повышение эффективности параллельного решения линейной задачи Коши на основе метода рекурсивного умножения матриц

Предложен масштабируемый параллельный метод матричного умножения на основе систолического и рекурсивного алгоритмов, который позволяет повысить эффективность решения линейной задачи Коши на основе экспоненциального метода. Для предложенного метода определено оптимальное значение глубины рекурсии и минимальное значение блока перемножаемых матриц. Разработаны схемы отображения метода на параллельные структуры с распределенной памятью и топологией решетка/тор.

I.A. Nazarova

The Rise of Efficiency of Parallel Decision of Linear Cauchy's Problem on the Basis of Method of Recursive Matrices Multiplication

The scalable parallel method of matrix multiplication is offered on the basis of systolic and recursive algorithms, which allows to promote efficiency of decision of linear Cauchy's problem on the basis of exponential method. For the offered method the optimum value of depth of recursion and minimum value of block of the multiplied matrices is certain. The schemes of reflection of method are developed on parallel structures with the distributed memory and topology of mesh/torus.

Стаття надійшла до редакції 10.07.2008.